A New Explicit Loss Notification with Acknowledgment for Wireless TCP

Wenqing Ding and Abbas Jamalipour
School of Electrical and Information Engineering
The University of Sydney
Sydney, NSW 2006, Australia

Abstract-**Due to commonly usage of the TCP/IP protocol stack in Internet applications that require reliable data transfer, it is important to keep the protocol stack (and also the network element structure) as unchanged as possible even when mobility features required in wireless Internet is added into the network. However, TCP performs poorly in wireless networks running at high bit error rates. In this paper, a new enhancement to TCP based on a previously proposed scheme is introduced. The method is based on the Snoop protocol and is similarly used for fixed host to mobile host direction. The proposed method has been compared with the Snoop protocol and other enhancement protocols proposed for the wireless TCP. It has been shown that the new explicit loss notification with acknowledgement has superior performance over Snoop and other protocols. The proposed method improves the end-to-end reliable transport performance in mobile environment.**

I. INTRODUCTION

Third generation wireless networks are going to start their service by the end of this year in Japan and in other countries in the following year. These 3G systems have been planned to use a packet core network (potentially, an IP core network) in contrast to the circuit-switched network used in second-generation systems. This packet core network has been planned in order to provide service to the emerging packet-based applications, and in particular the Internet applications, more efficiently while still it can handle voice and other circuit-based applications, e.g. by means of voice over IP techniques. 3G systems promised higher data bit rates around 384 kbps to mobile users and up to 2 Mbps to indoor mobile hosts than their 2G counterparts.

Considering the dominant data traffic services promised in 3G systems and its connectivity with the wired network, we could expect significant usage of the commonly used TCP/IP protocol stack in these systems. This is because of the fact that many popular Internet applications such as electronic mail, file transfer, web browsing, and remote network access, require a reliable data transfer, as the one provided by TCP, and that the vastly deployed wired Internet around the world has already adapted TCP/IP as its main protocol stack. In most situations, a mobile host will have some type of communications with a fixed host connected to the wired part of the network (Fig. 1). Thus any extension of the wired Internet into wireless environment should consider usage of a similar protocol stack and also minimal change in the software deployed in the fixed hosts.

Mobile IP, defined in IETF RFC 2002 [1], tries to provide mobility feature for the wireless and mobile Internet at network layer. It modifies the, usually fixed, IP address of a host connected to the Internet into a virtual address form, namely care-of address, so that the mobile host can move around and change its point of attachment to the network without violating the IP address configuration of the Internet and still maintain its connection. TCP, the transport control protocol of the TCP/IP protocol stack, though does not show any modification requirement when the fixed network changes to a heterogeneous wired/wireless network, could actually implicate the performance of the Internet connectivity in wireless networks.

TCP is the most commonly used protocol at the transport layer of the network stack in the Internet, originally developed for wired networks with low bit error rate (BER) in the order of less than 10^{-8} [2]. In this context, any wireless network with Internet service needs to be compatible with the protocol used in the wired network, i.e. the TCP/IP protocol. There are however, some design issues in TCP protocol, which make it difficult to be used efficiently over the wireless and satellite links. Recently, there have been vast research activities comparing the performance of TCP in high-BER and high-latency channels (e.g. see [3-4]) and modification proposals to improve its performance in terrestrial cellular and satellite wireless networks [5-10].

In this paper, we will discuss the inefficiency of TCP for heterogeneous wired/wireless networks and introduce a new scheme to improve the TCP performance in those networks. We will compare our proposed method with other methods already proposed to alleviate this inefficiency and show that how the proposed method in this paper can achieve superior performance.

In the following section, we will review the most important modifications to TCP that have been proposed to improve TCP performance in wireless environment. In Section III, we will introduce the concept of the new explicit loss notification with acknowledgment scheme and its implementation procedure will be discussed. In Section IV, we will explain the simulation used in this paper and discuss the numerical results on comparison of the new scheme with other schemes. Finally, Section V, concludes the paper.

0-7803-7244-1/01/$10.00 ©2001 IEEE

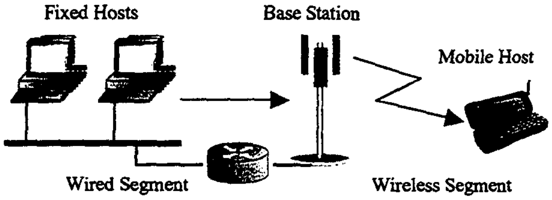

Fig. 1. TCP packet flow within a wired/wireless network.

II. RELATED WIRELESS TCP SCHEMES

TCP has been designed and tuned for networks in which segment losses and corruptions are mainly due to network congestion. This assumption might be invalid in many of the emerging networks such as wireless networks. The flow control mechanism used in TCP is based on time out and window-size adjustment, which can work with high utilization in wired networks with low BER in the order of 10^{-8}. However, when the wireless channel is used (partially or totally) as the physical layer with a BER as high as 10^{-3}, it may perform inefficiently. The reason is that in the wireless channels the main cause for packet loss is the high BER and not the network congestion, which was assumed for wired networks. The low efficiency of the TCP in wireless channel is a direct result of the fact that the TCP misinterprets the packet loss because of high error rate and that of congestion. On the other hand, in high-latency networks (such as cellular and satellite networks) adjustment of the window size could take a long time and reduce the system throughput.

TCP has the ability to probe unused network bandwidth by a mechanism called *slow start* and also to back off the transmission rate upon detection of congestion through the *congestion avoidance* mechanism [2]. At the connection startup, TCP initializes a variable called *congestion window* to a value of one segment. This variable determines the transmission rate of TCP. The window size is doubled at every round trip period and then increases linearly until a packet loss is experienced. At this time, the congestion avoidance phase is commenced, the window size is halved, and the lost packet is retransmitted. During this phase of TCP, the window size is increased only linearly by one segment at each round trip period and might be halved again upon detection of another packet loss. If the retransmitted packet is lost, the timeout mechanism employed in TCP reduces the window size to one. Since all these procedures are performed at the periods equal to round trip delay of the channel, the system throughput could be degraded significantly where high-latency channels are involved.

There are some modifications to the basic TCP that can be made in order to perform more efficiently in high-latency and error-prone wireless networks with Internet services. *Selective acknowledgment* (SACK) TCP [9] for example is a method in which multiple losses in a transmission window can be recovered in one round trip period instead of two in the basic TCP. *TCP for transactions* (T/TCP) also reduces the user perceived latency to one round trip delay for short transmissions. In *TCP spoofing*, a router close to the base station is considered, which sends back acknowledgments for the TCP data. The responsibility of any segment loss in this method comes to the router. In another method, called *split TCP* [7-8], a TCP end-to-end connection is divided into multiple TCP connections (wired and wireless types) and a special *wireless TCP* connection is employed for the wireless link part. The method, which sometimes called Indirect-TCP, tries to separate packet loss over wired and wireless parts of the link (commonly occurred because of different reasons, i.e. congestion versus bit error rate) but asks for having kind of violation in the semantics of the TCP as an end-to-end protocol.

Link layer protocols are another alternative for improving the poor performance of TCP over wireless link [5]. In those methods usually forward error correction (FEC) or automatic repeat request (ARQ) methods are used to improve the performance. Independent timer reaction at link and transport layers that may result in unnecessary retransmissions, fast retransmission interaction, and large round-trip variations are considered as major problem with link-layer approaches [10].

The last enhancement to TCP for wireless channel that we review here is called Snoop protocol [6]. In this method, the base station is equipped with a module called snoop agent, which its function is to monitor the TCP packets transmitted from a fixed host to a mobile host and vice versa. The agent caches all those packets locally and in the case of receiving duplicate acknowledgments (ACKs), retransmits the packets promptly and suppresses duplicate ACKs. The Snoop protocol performs retransmission of lost packets locally (at the base station) and hence avoids lengthy fast retransmission and congestion control at the sender side. By this method, end-to-end semantics of TCP is maintained and performance of TCP is improved. The Snoop protocol is mainly used for the fixed host to mobile host direction, though an explicit loss notification algorithms complementing the Snoop on the mobile host to fixed host direction has been later proposed in [10]. A comprehensive comparison between the methods proposed for improving TCP performance over wireless channel is given in [3].

III. EXPLICIT LOSS NOTIFICATION WITH ACKNOWLEDGMENT SCHEME

The poor performance of TCP in error-prone wireless networks is mainly due to lack of explicit information at the transport layer on the reason of a packet loss. This type of information was not required when TCP has been developed since TCP has been designed to work in wired networks with low bit error rates and where the main reason for the packet loss was network congestion. Other unusual reasons in those networks could be ignored without any difficulty in operation of window-based and timeout-based TCP. For the wireless networks, if we can explicitly inform TCP the reason of a packet loss, then TCP will be able to maintain its throughput (i.e. not to reduce its congestion window size) if the packet has been lost not because of network congestion.

The methods reviewed in the previous section tried to improve the performance of TCP in wireless networks. But none of these algorithms actually lets TCP sender know clearly whether the packet is lost due to wireless error or network congestion. This makes the TCP sender retransmits the packet as usual (or quicker than usual) and then cannot keep the throughput high in the error prone environment.

Snoop protocol [6] is a good scheme to improve the performance of TCP in wireless network at fixed host to mobile host direction. But the Snoop protocol retransmits the lost packet like other link layer solutions, now locally but through its snoop agent. The Snoop protocol also suffers from not being able to completely shield the sender from the wireless losses.

Based on Snoop protocol, we proposed a new protocol called Explicit Loss Notification with Acknowledgment (ELN-ACK), which can remedy limitations of the Snoop protocol. In ELN-ACK protocol implementation, we need to make modifications to the structure of acknowledgment packet, and the software part at base station, mobile host and fixed host. Those modifications, however, can be maintained at minimal compared with other schemes. The method still looks at the throughput and delay performance improvement of TCP at the fixed host to mobile host direction (Fig. 1).

A. *New Acknowledgment Packet*

In ELN-ACK a new form of acknowledgment packet called ACK_{ELN} is used. The sequence numbers of the four most recently lost packets judged by the mobile host and one bit (called ELN bit) to indicate the reason of the lost packet, are included in the ACK_{ELN} acknowledgment packet. A '1' in the ACK_{ELN} indicates the packet is lost in the wired network congestion and a '0' in the ACK_{ELN} indicates the packet is lost due to wireless error. Therefore, the reason of a packet loss is explicitly informed to the TCP sender. The default value of the ELN bit transmitted by a mobile host is "1" (i.e., assuming the corresponding packet was lost due to network congestion).

The ELN bit is judged at the base station. ELN agent at the base station (to be introduced shortly) checks the information stored in the ELN bit to see if the packet has been lost before it is arrived at the base station. If the agent found the packet had lost before it arrived at the base station, it remains the corresponding ELN bit to "1,' otherwise it will change the ELN bit to '0.' After the ACK_{ELN} is processed by the ELN agent at base station, it continues to transmit back to fixed host. When the fixed host (the original sender) receives the ACK_{ELN}, the TCP sender will know the reason of packet loss from the ELN bit explicitly. A flowchart summarizing the ACK_{ELN} processing is shown in Fig. 2.

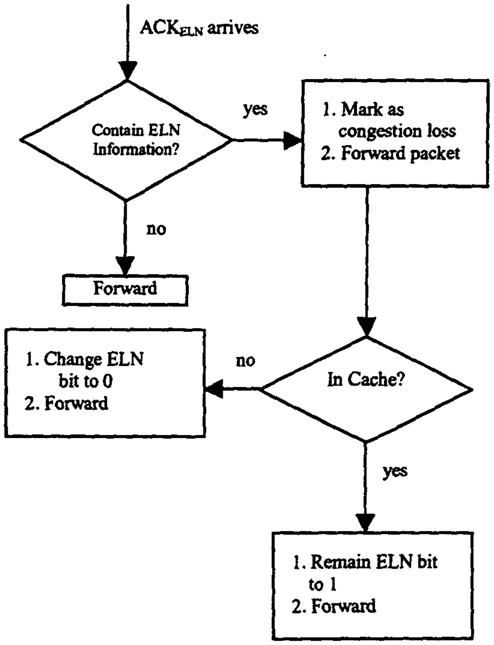

Fig. 2. Flowchart for ACK_{ELN} processing in ELN-ACK.

B. *ELN-ACK Agent at the Base Station*

Similar to the snoop agent used in the Snoop protocol [6], an ELN-ACK agent is introduced at the base station that has two main functions. One is to judge and store the packet loss information transmitted from the fixed host. Like ordinary wired networks, packet transmitted from the fixed host to base station may be lost due to congestion. If the base station receives a packet, which is not in sequence, it will store the corresponding packet information in the ELN-ACK agent. Using the stored information, the base station can judge the reason of packet loss when it receives the ACK_{ELN} transmitted from mobile host. The second function is to judge the value of ELN bit. When the base station receives an ACK_{ELN}, it will judge the lost packet based on the stored information in the acknowledgment packet. If it finds that the packet has been lost before arriving at base station, it will fill the ELN bit with '1' to indicate the packet was lost due to congestion. If the lost packet has already arrived at the base station, it fills the ELN bit with '0' to indicate the packet was lost in the wireless channel. Data processing procedure at the ELN-ACK agent is very similar to the one used in Snoop protocol [6].

C. *TCP Sender Procedure*

When the fixed host receives the ACK_{ELN}, it actions with the information stored in the ELN bit. If the ELN bit is '1,' it means that the corresponding packet is lost due to wired segment congestion and thus it will proceed with the same procedure as in the window algorithm. If the ELN bit is '0,' it means that the corresponding packet is lost due to wireless error and thus it retransmits the packet immediately without any window reduction.

IV. SIMULATION AND PERFORMANCE RESULTS

We performed several experiments to measure the perfomance of data transfer from fixed host to mobile host. Figure 3 shows a simple network used for the simulations in this paper to send TCP packets from a fixed host to a mobile host. The base station includes a finite-buffer drop-tail gateway, and the network has wired and wireless links. ELN-ACK protocol has been implemented using C++ programming and Network Simulator (NS-2) simulation package [11] has been used to simulate the TCP packet transmission in wired/wireless segments of the network. In the simulation, some parameters can be set to indicate different network conditions. Those parameters are summaried as in following:

1. Buffer size (*B*, packets) in the base station,
2. Propagation delay (*D*, msec) which includes: a) the time between the release of a packet from the source and its arrival into the link buffer; b) the time between the transmission of the packet on the bottleneck link and its arrival at its destination; and c) the time between the arrival of the packet at the destination and the arrival of the corresponding acknowledgment at the source,
3. The bandwith (*U*, packets/msec) of bottleneck link from base station to mobile host.

The throughput performance (i.e., the total number of original packets received by the receiver in a given period of time) of ELN-ACK, TCP Reno, TCP Tahoe, TCP Snoop, TCP Sack, TCP-Split have been investigated under different network conditions. The simulation results are shown in Figs. 4 through 6.

In the simulation, first packet sequence trace of a bulk TCP transfer using the proposed ELN-ACK and conventional (wired-optimal) TCP Reno has been considered. The simulation was performed under a typical wireless packet loss rate of 0.05 [12]. We find that in TCP Reno, when wireless errors occur, TCP throughput performance degrades significantly. In particular, we have observed that a coarse packet loss rate of about 5% led to a throughput performance degradation of a factor of 4.5 from ideal. A closer analysis of the packet sequence trace reveals the reasons for this. During the course of the transfer, packets are lost over the wireless link and need to be retransmitted. Every time the TCP sender detects the loss of a packet, it retransmits the lost packet, but also reduces its congestion window, ascribing the loss to be due to network congestion. This is the correct interpretation for wired networks, but is usually an incorrect response in networks that have wireless links. As a result, the average TCP window size is small and timeouts are frequent. A stairs-like packet transfer over the time occurs in the process of the connection in TCP Reno due to those timeouts, which leads to degraded peformance.

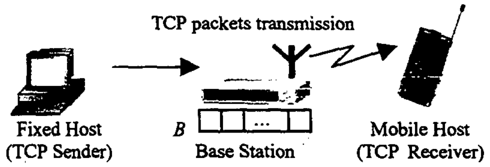

Fig. 3. Network topology used in the simulation.

Figure 4 shows the throughout performance of TCP Reno, TCP Tahoe, TCP Split, TCP SACK, TCP Snoop, and the proposed TCP with ELN-ACK under condition that there is a buffer with size of 5 packets at the base station and the propagation delay is 0.2 msec. The link bandwidth is set to 100 packets per msec for this simulation. Based on the results shown in this figure, the Snoop and ELN-ACK protocols provide significant improvement when the packet loss rate becomes larger than around 0.3%. The throughput performance of the Snoop and ELN-ACK protocols however, remains very close until packet loss rate of 1% but after that, the ELN-ACK performs better than the Snoop protocol. The better performance of Snoop and ELN-ACK compared with other TCP methods is clear since these two protocols provide better differentiation of the packet loss types over wireless link (bit error-related packet loss) and the that of wired link (congestion-related packet loss). However, the ELN-ACK protocol improves the throughput performance even more by sending information on the reason of packet loss to the TCP sender whereas the Snoop protocol tries to handle all wireless-related losses at its snoop agent located in the base station. In other words, the ELN-ACK protocol adds extra features to the Snoop protocol and immunes all packet loss even when the packet loss rate is high and the snoop agent cannot handle them. This is the reason of better performance of ELN-ACK compared with the Snoop protocol in high error bit rates.

Figure 5 compares the throughput performance of the above TCP protocols with the same conditions as used in Fig. 4 but when the link has a smaller bandwidth. In this situation, packet contention rate is higher and thus the congestion-related packet loss becomes more important and therefore the performance improvement of Snoop and ELN-ACK protocols is exhibited at higher packet loss rates compared to Fig. 4.

With the same link bandwidth but with larger buffer size at the base station, Fig. 6 shows the throughput performance of different TCP algorithms. In this situation, still ELN-ACK performs the best compared with other schemes.

From simulation shown in Fig. 4 through 6, we can see that the throughput of TCP Reno and TCP Tahoe drop sharply when the packet loss rate is increased to above 10^{-2}. Throughput of TCP Reno and TCP-Tahoe drop to only 10%~20% compared with error free wireless link whereas the ELN-ACK scheme can keep the throughput as high as 80%~90% of error free environment. There are significant performance benefits of using the ELN-ACK protocol. The main advantage of ELN-ACK is that it helps maintianing a large TCP congestion window at high wireless error rate.

V. CONCLUSIONS

In this paper, we proposed a novel protocol named Explicit Loss Notification with Acknowledgment (ELN-ACK) based on Snoop protocol to improve the transport performance from the fixed host to mobile host. The key idea here is to let the fixed host of TCP sender knows clearly of the reason of the packet loss in a wireless network; that is whether it is a congestion-related packet loss or a wirless error-related one. Simulation results have shown that the new protocol suite is robust and can significantly improve wireless link throughput, and the improvement for high-speed wireless links is more significant when loss rate is high in the wireless link. The delay performance of the new ELN-ACK technique and its comparison with other TCP techniques is an undergoing research and the results will be published in another paper.

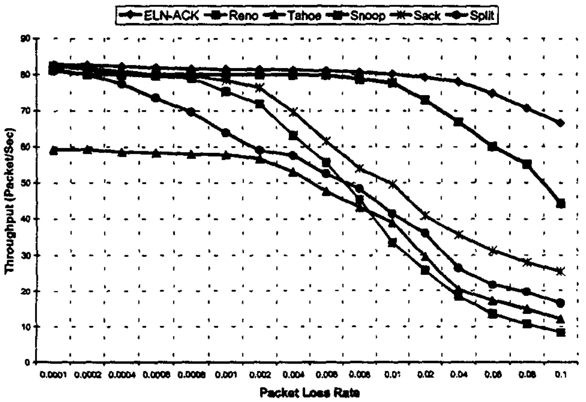

Fig. 4. Throughput comparison for B = 5 packets, D = 0.2 msec, and U = 100 packets/msec.

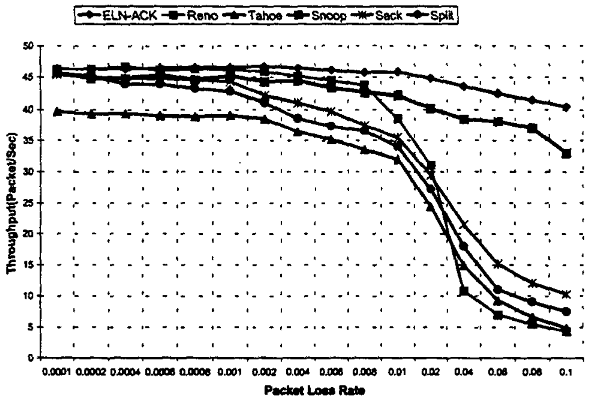

Fig. 5. Throughput comparison for B = 5 packets, D = 0.2 msec, and U = 50 packets/sec.

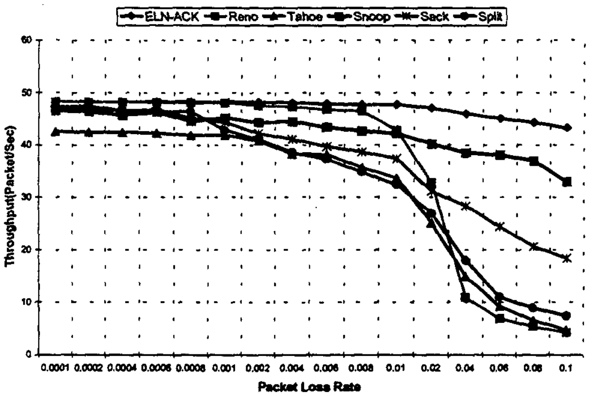

Fig. 6. Throughput comparison for B = 8 packets, D = 0.2 msec, and U = 50 packets/msec.

REFERENCES

[1] C. E. Perkins, *Mobile IP-Design, principles and practice*, Reading, MA: Addison Wesley Longman, 1998.

[2] W. R. Stevens. *TCP/IP Illustrated*, Volume 1. Addison-Wesley Longman, MA, 1994.

[3] H. Balakrishnan, V. N Padmanabhan, S. Seshan, and R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," IEEE/ACM Trans. On Networking, vol. 5, no. 6, pp. 756-769, December 1997.

[4] K. Fall and S. Floyd, "Simulation-based comparisons of Tahoe, Reno, and Sack TCP," Computer Communication Review, vol. 26, no. 3, pp. 5-21, July 1996.

[5] E. Ayanoglu, S. Paul, T. F. LaPorta, K. K. Sabnani, and R. D. Gitlin, "AIRMAIL: A link-layer protocol for wireless networks," ACM ACM/Baltzer Wireless Networks Journal, vol. 1, pp. 47-60, February 1995.

[6] H. Balakrisnan, S. Seshan, and R. H. Katz, "Improving reliable transport and handoff performance in cellular wireless networks," ACM Wireless Networks, vol. 1, no. 4, pp. 469-481, December 1995.

[7] A. Bakre and B. R. Badrinath, "Handoff and system support of indirect TCP/IP," Proc. Second Usenix Symp. on Mobile and Location-Independant Computing, April 1995.

[8] A. Bakre and B. R. Badrinath, "I-TCP: Indirect TCP for mobile hosts," Proc. 15th International Conf. On Distributed Computing Systems (ICDCS), May 1995.

[9] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP selective acknowledgment options," RFC-2018, 1996.

[10] H. Balakrishnan and R. H. Katz, "Explicit loss notification and wireless web performance," IEEE Global Telecommunications Conference (Globecom'98), Mini Conference, Sydney, Australia, Novermber 1998.

[11] NS-2 simulation tool home page. http://www.isi.edu/nsnam/ns/, 2000.

[12] J-H Hu, K. L. Yeung, S. C. Kheong, and G. Feng, "Hierarchical cache design for enhancing TCP over heterogeneous networks with wired and wireless links," Proc. IEEE Global Telecommunication Conference (Globecom2000), San Francisco, CA, Nov. 27 - Dec.1, 2000.